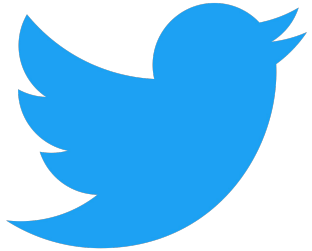
Ruby on Rails

Introducción

Historia de Rails

- Rails es un framework para desarrollar aplicaciones web dinámicas.
- Creado en 2004-2005 por David Heinemeier Hansson

Quién usa rails?



Por qué usar rails?

Convention over Configuration

- Menos código que escribir
  - A veces rails ya genera código por nosotros.
  - A veces no hay ningún código que escribir porque se asume un comportamiento.
- Otro beneficio de este patrón:
  - Se aprende una vez
  - Se aplica el concepto aprendido en el siguiente proyecto.
  - Se puede saber qué esperar!

Abstracción de base de datos

- No hay necesidad de interactuar con la base de datos.
- Active Record actúa como un ORM
- Es importante entender el SQL Generado

Otras características

- Agile-friendly
- Principio: Don’t Repeat Yourself (DRY)
- Multiplataforma
- Open Source
- Modular:
    - Permite que cambiemos los componentes que realizan un trabajo similar.

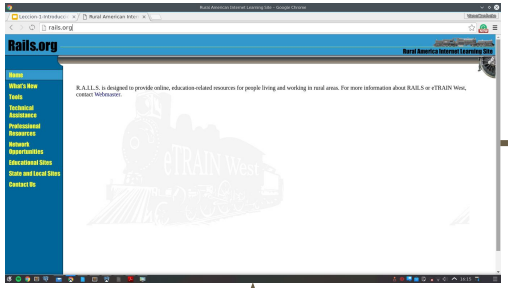
Utiliza SQLite por defecto

- Autocontenido
- Sin necesidad de tener un servidor
- Cero - configuraciones
- Transaccional
- Motor de base de datos relacional

MVC: Modelo Vista Controlador

- Inventado en 1979 por Trygve Reenskaug
- Un patrón de software bien establecido utilizado por muchos frameworks web y desktop.
- **Model**: representa los datos con los que está trabajando la aplicación.
- **View**: representación visual de un dato (htlm, json, js)
- **Controller**: orquesta la interacción entre el modelo y la vista.
    - Se podría tener una capa de modelo que obtenga los datos de la base de datos.
    - Se podría tener un controller que defina cómo se va a mostrar el dato: html, json, xml, etc.

Ciclo MVC



CONTROLLER

VIEW

MODEL

1. Request
2. **Controller** ⟷ **Model**
3. **Controller** invoca **View**
4. **View** despliega datos

Entonces...

- **Rails** es bueno para un prototipado rápido
- **MVC** y Convention over Configuration permite a los desarrolladores “Pensar más y hacer menos”, simplemente sigue los patrones!